TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
MECHANICS | MECHANIKA
2-M/2015

SZYMON TENGLER, ANDRZEJ HARLECKI*

COMPUTER PROGRAM FOR DYNAMIC ANALYSIS OF SPECIAL VEHICLES WITH HIGH GRAVITY CENTRE

PROGRAM KOMPUTEROWY DO ANALIZY DYNAMIKI SAMOCHODÓW SPECJALNYCH O WYSOKO POŁOŻONYM ŚRODKU CIĘŻKOŚCI

Abstract

The functionality of the authors' computer program which enables to analyze dynamics of special cars with a high gravity center, in particular, a technical rescue vehicle of fire service is presented in this article. The correctness of the analysis has been confirmed experimentally. The program consists of two essential parts: the calculation module and the animation module designed for three-dimensional visualization. Properly prepared input data in the XML format is processed by the calculation module which can solve the equations of motion including an uneven road surface. The results of this module are processed by the animation module performing a smooth computer animation. For mathematical modeling of road unevenness and presenting three-dimensional animated objects authors used an unconventional combination of the developed program with the Blender graphical environment.

Keywords: *computer modeling, dynamic analysis, special vehicles*

Streszczenie

W artykule przedstawiono funkcjonalność autorskiego programu komputerowego umożliwiającego analizę dynamiki samochodów specjalnych o wysoko położonym środku ciężkości, a w szczególności samochodów ratowniczo-gaśniczych straży pożarnej. Poprawność analizy potwierdzono doświadczalnie. Program zbudowano z dwóch zasadniczych części: modułu obliczeniowego i modułu animacji służącego do wizualizacji trójwymiarowej. Odpowiednio przygotowane dane wejściowe w formacie XML są przetwarzane przez moduł obliczeniowy rozwiązujący równania ruchu z uwzględnieniem nierównej nawierzchni drogi. Wyniki tego modułu są przetwarzane przez moduł animacyjny umożliwiający wykonanie płynnych animacji komputerowych. Do matematycznego modelowania nierówności nawierzchni drogi i prezentacji animowanych obiektów trójwymiarowych autorzy wykorzystali niekonwencjonalne połączenie opracowanego programu ze środowiskiem graficznym Blender.

Słowa kluczowe: *modelowanie komputerowe, analiza dynamiki, samochody specjalne*

* Ph.D. Eng. Szymon Tengler; Ph.D. D.Sc. Eng. Andrzej Harlecki, prof. ATH; Faculty of Mechanical Engineering and Computer Science, University of Bielsko-Biala.

1. Introduction

Computer simulations of a motion of vehicles mentioned in the abstract, performed in the case of different variants of constraints, can predict a lot of potential threats, and the conclusions drawn from them can constitute important guidelines for vehicle designers. It is important for the prepared mathematical model of the vehicle to be a sufficiently accurate representation of the real system. According to the vehicle designers the computer simulations of an appropriately prepared mathematical model shorten the time of testing on a real prototype of the vehicle, and also reduce costs of the designing process.

Algorithms for generating the equations of motion have been formulated by homogenous transformations and Lagrange's equations [3]. Generating these equations takes into account different types of experiments, e.g.: changing a lane plane, driving on a curved track, driving over uneven road surfaces, negotiating a circular track (driving on the track in the shape of a circle). Both the speed of the vehicle and its path can be controlled by a PID controller.

2. Computer program

The devised computer program was divided into two modules: a solver module and an animation module.

2.1. Solver module

The developed solver module consists of mutually co-operating elements, which are specified in a diagram of the components (Fig. 1).

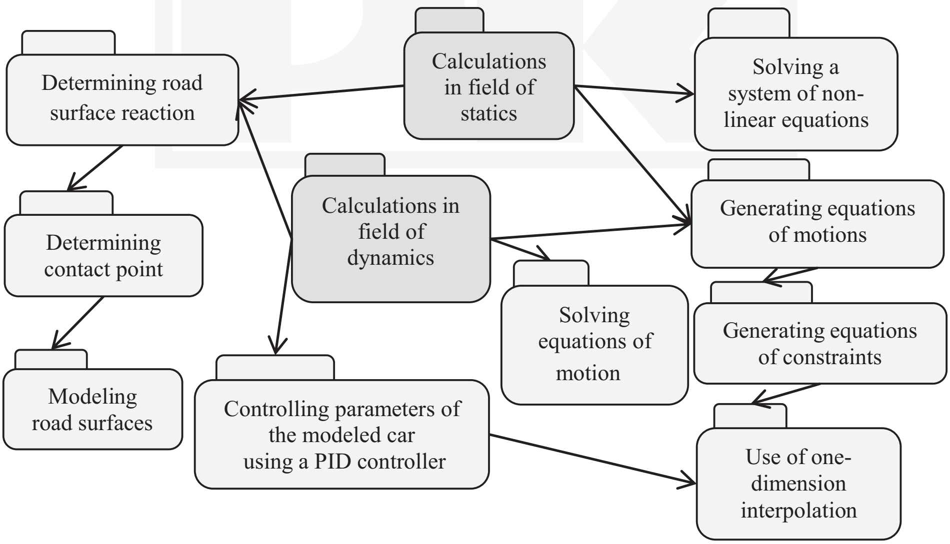

Fig. 1. A diagram of the components of the solver module

The main role is performed by the components used for making calculations from the field of statics and dynamics. The arrows guided appropriately symbolize their co-operation with other components of the solver module, which are characterized in Table 1.

Table 1

Characteristics of the solver module components

Name of component	Description
Solving system of non-linear equations	It serves to solve equations of statics [3] by the iterative Newton's method
Determining reaction of a road surface	It implements three models of tires: Fiala, Pacejka and Dugoff-Ufelmann [9]
Determining contact point	It implements algorithms which allow to determine the position of the point in which road surface reactions are applied to vehicle wheels
Modeling road surface	It implements a continuous and discrete model of the road surface described in the work [10]
Generating equations of motion	It generates equations of motion
Generating equations of constraints	It generates the so called equations of constraints described in the work [9]
Use of one-dimension interpolation	It implements functions of splines of the first and third order
Solving equations of motion	It implements methods of solving differential equations: fixed-step (Runge-Kutta IV order, Implicite Eulera) and variable-step: (Runge-Kutta-Fehlberg, Bulirsha-Stoera-Daufhardta). Advantages and computing power of the methods mentioned are presented in the work [8]
Controlling parameters of modeled vehicle by use of PID controller	It implements the operation of a PID controller

The component which controls parameters of the modeled vehicle in accordance with the PID controller algorithm should be distinguished among the components presented in table 1. Its operation is used to provide the determined course of vehicle speed values or to maintain the trajectory of its motion in a shape of a circle. The initial value (controlling) of the PID controller depends on the reference value. In the first case the reference value is an assumed course of speed values of the modeled vehicle gravity centre (given e.g. in the form of a spline function), and the initial value is the course of values of driving torques applied to the rear wheels. In the second case the reference value is the assumed radius of a circular track, and the initial value is a turning angle of the front wheels. A method of manual adjustment in connection with the Ziegler-Nichols method was used to determine the PID controller amplification [12].

The solver performs calculations on the basis of property prepared input data obtained from the pre-processor. The input data contains all the necessary parameters needed for the proper operation of the solver module (Fig. 1). These parameters, among others, are:

- the type of analysis (statics and/or dynamics),
- simulation time and the type of a method solving equations of motion,
- parameters of bodies of a multibody system,
- parameters of equations of constraints,
- coordinates of mounting points of spring and damping elements,
- a stiffness coefficient of elastic and damping elements,
- parameters needed to control the vehicle speed value (PID),
- parameters needed to maintain the track of vehicle motion in the shape of a circle,
- parameters of particular tire models ,
- parameters of a road surface model,
- initial conditions (the vehicle parameters at the initial moment $t = 0$).

The mentioned parameters' data is transferred to the solver module in format XML (eXtensible Markup Language) [7]. On the basis of the prepared data the solver module performs calculations.

2.2. Animation module

The second element of the program is the three dimensional animation module. Its main components are presented in Fig. 2.

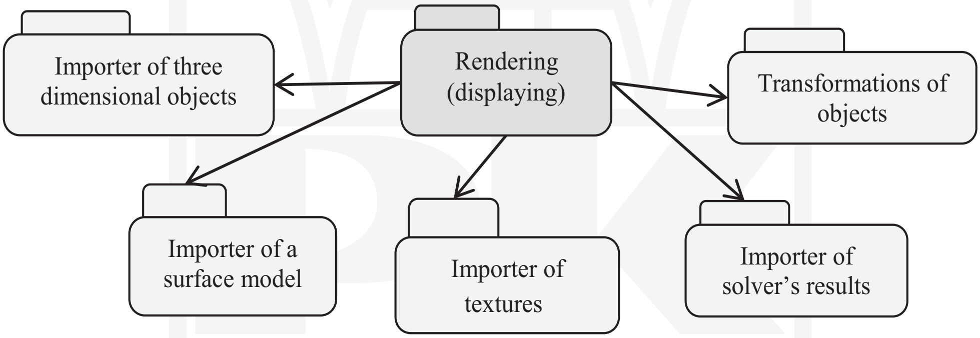

Fig. 2. A diagram of the components of the three dimension animation module

The components for importing three dimensional objects and textures are made in the form of libraries dll in language C++ [1]. The main joining element (the Rendering component) and other components co-operating with it are implemented in NET technology while using C# language [4] and OpenGL library [11].

The importer of objects enables to import any three dimensional objects written in the 3ds format. 3ds Max Autodesk company and Blender are popular environments for creating objects in this format on the information market. The first of the tools is commercial and the other is free of charge. However, for the needs of this work the Blender environment was selected because it can be used free of charge. Examples of the screen shots, presenting the selected model of the component (body building of a vehicle) made in the Blender environment, and then imported to the animation module, are presented in Fig. 3.

a) b)

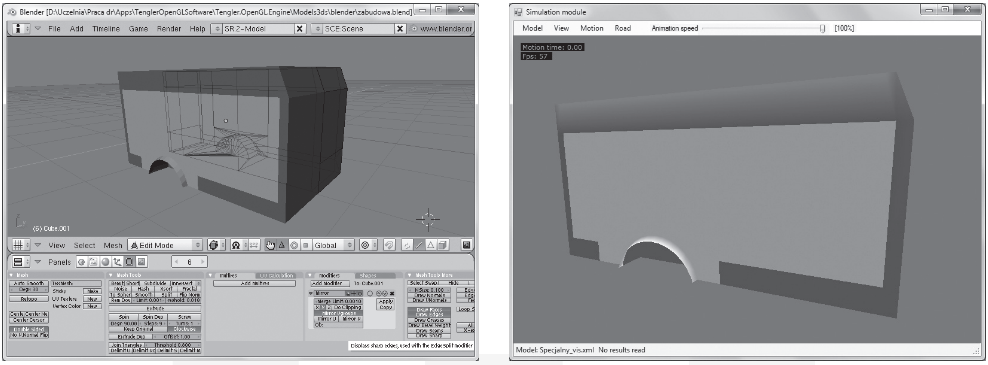

Fig. 3. Examples of screen shots showing operating of the component for importing 3ds objects: a) a 3ds object made by use of the Blender program, b) a 3ds object imported to the animation module

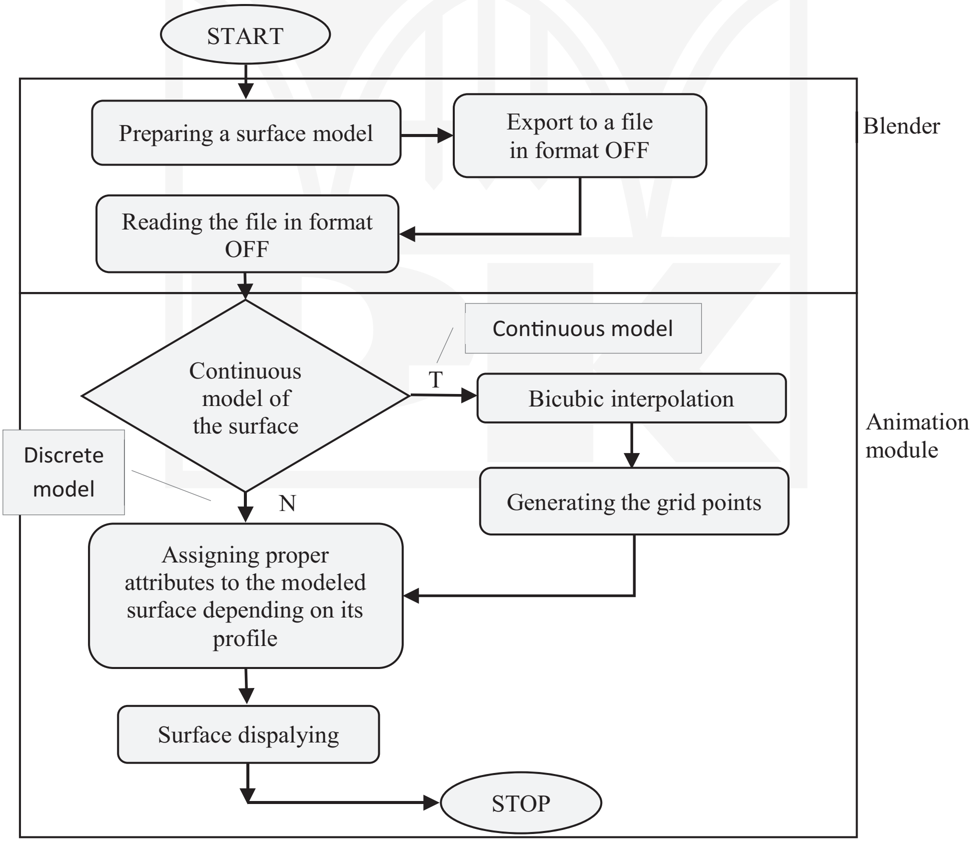

Fig. 4. A chart of operations concerning the rendering process of the road surface

The Blender program was also used for modeling different types of road surface. Import of the surface model from the Blender environment to the animation module is a slightly more complicated process than the import of the 3ds objects. It takes place according to the stages illustrated in Fig. 4.

A road surface model is prepared in the first step. Then, it is exported to a file in format OFF (Object File Format) [6]. Both of these operations are made in the Blender environment. The exported OFF file contains, among others, a set of coordinates of vertices (points) making a surface model. Depending on the selection of the surface model these points are used in the process of road surface rendering in an appropriate way. When computer simulations concern driving a vehicle over the modeled surface in a continuous way, then it is rendered by the use of the Bicubic interpolation method. An example of the model surface made by program Blender, and then imported to the animation module, is presented in Fig. 5.

a) b)

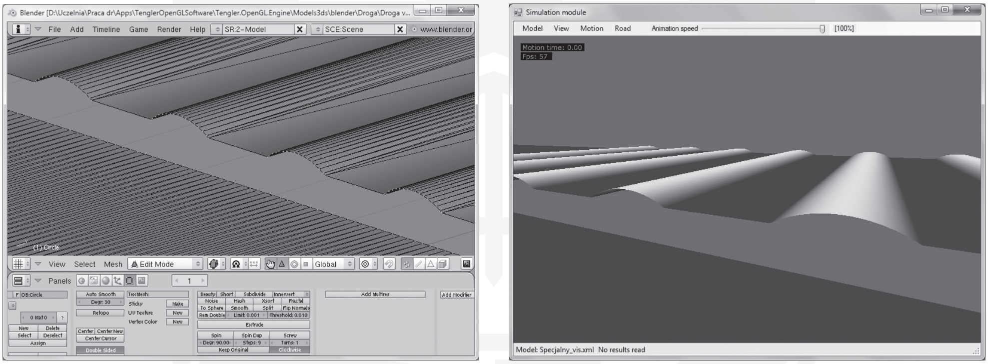

Fig. 5. An ex ample of the surface model made in the Blender environment a) made by program Blender b) imported to the animation module

Other components of the animation module are characterized in Table 2.

Table 2

Characteristics of the animation module components

Name of a component	Description
Importer of textures	The component developed for importing graphic files in formats jpg, png, bmp and tiff, used in the process of imposing textures on the animated objects
Transformations of objects	The component developed for translations and rotations of the animated objects
Importer of the solver module results	The component importing the results generated by the solver module. The imported results are interpolated by a spline function of the first order to provide smoothness of animation

Particular components (Fig. 2) were programmed, using a technique of object-oriented programming with the application of design patterns [2]. A diagram of the classes of the Rendering component, presenting its main functionality, is shown Fig. 6.

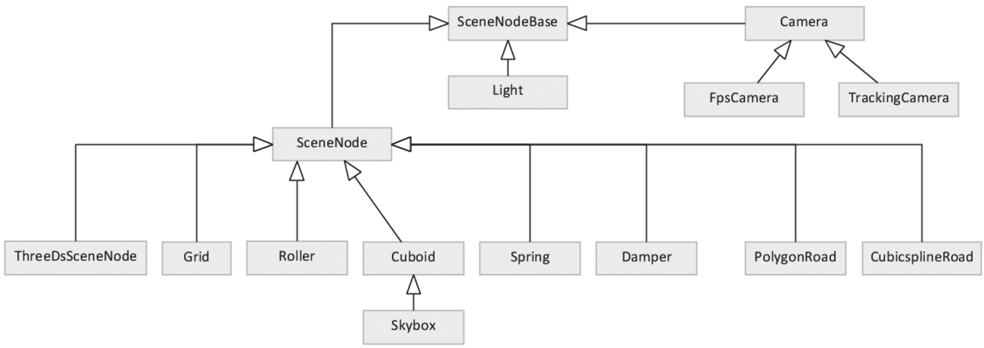

Fig. 6. The basic diagram of the rendering component classes

The base class for all objects is an abstract class – SceneNodeBase. From it the following classes inherit directly:

– Light – the class responsible for "scene" light,
– Camera – the class providing an appropriate view of objects on the scene; two independent cameras are available: FpsCamera (First Person Shooter) and TrackingCamera following the indicated object [5],
– SceneNode – the base class for the objects displayed on the "scene".

Other classes illustrated in Fig. 6, inheriting from the SceneNode type, are the basic rendering elements of the characterized module.

a) b)

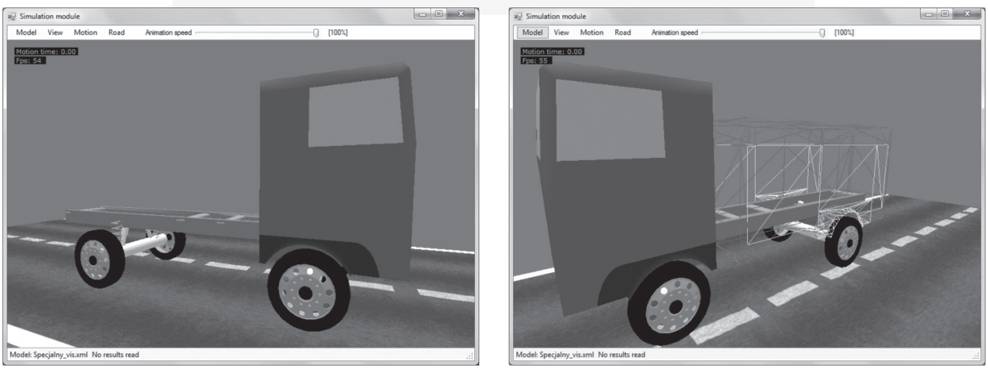

Fig. 7. The examples of the screen shots presenting: a) the hiding of modeled components, b) a "skeleton" view of the components

The presented structure of the prepared animation module has a universal character and enables to extrapolate it in a relatively easy way. With little effort it can be used e.g. for an animation of components, using the simulation results which come from other computing packages. Owing to the component for importing 3ds objects those can be animations of any complex shape.

The described animation module also offers a lot of useful functions connected with handling the animation process. A camera can be adjusted here in a lot of ways in order to follow "easily" the moving objects. Some objects can be also "hidden" to find out how particular modeled components of a vehicle move (Fig. 7).

3. Conclusions

A structure of two basic modules of the developed program to analyze dynamics of special vehicles with high gravity centre, that is the solver module and the animation module, is described in the article. A fact of a rarely met connection of authors' computer program with the Blender environment should be emphasized. Owing to this, in this environment both road surface models and three dimension objects of any shapes, which then are imported in the authors' program, can be built. Such a connection, according to the authors, shortens the time to prepare the final virtual model and allows to perform computer simulations in an effective way. The particular modules were made while using different programming languages and techniques of object-oriented programming combined with design patterns. An intuitive language of XML tags was selected as a format for transferring data between the modules. According to the authors the use of the object-oriented programming technique enables to expand the developed computer program with additional functionalities in a relatively easy way by other programmers. Additionally, the use of the XML language as "a bridge" between the developed modules, does not only allow to make separate programs co-operating with the solver module, but also to use the animation module for three dimension visualization of results, which come from other computing programs.

References

[1] Eckel B., *Thinking in C++*, Helion, Gliwice 2002.

[2] Gamma E., *Wzorce projektowe − inżynieria oprogramowania*, Wydawnictwa Naukowo-Techniczne, Warszawa 2005.

[3] Grzegożek W., Adamiec-Wójcik I., Wojciech S., *Komputerowe modelowanie dynamiki pojazdów samochodowych*, Wydawnictwo Politechniki Krakowskiej, Kraków 2003.

[4] Jeffries J., *Programowanie ekstremalne w C#*, APN PROMISE, Warszawa 2005.

[5] Pallister K., *Game programming Gems 5*, Charles River Media, Inc., Hingham, Massachusetts 2005.

[6] Rost R.J., Bourke P., *OFF − A 3D Object File Format* (http://paulbourke.net/dataformats/off).

[7] Rusty E., Scott W., *Means XML in a Nutshell*, O'Reilly, 2004.

[8] Szczotka M., Tengler S., Wojciech S., *Numerical Effectiveness of Models and Methods of Integration of the Equations of Motion of a Car, Differential Equations and Nonlinear Mechanics*, Hindawi Publishing Corporation, Vol. 2007, Article ID 49157, 2006.

[9] Tengler S., *Analiza dynamiki samochodów specjalnych o wysoko położonym środku ciężkości*, Praca doktorska, Wydział Budowy Maszyn i Informatyki, Akademia Techniczno-Humanistyczna, Bielsko-Biała 2012.

[10] Tengler S., Harlecki A., *Ways of uneven road surface modeling used in the vehicle dynamics analysis*, Journal of KONES, Vol. 19, No. 1, 2012.

[11] Wright R.S, Lipchak B., *OpenGL. Księga eksperta*, Helion, Gliwice 2005.

[12] Ziegler J.G., Nichols N.B., *Optimum Settings for Automatic Controllers*, Transactions of the ASME, Vol. 64, 1942, 759-768.